

CLUBE

Z 80

ABRIL /83

N.º 7

CLUB



N E S T E N Ú M E R O

PASSO A PASSO - O "B-A-BA" DO BASIC	2
INTRODUÇÃO À LINGUAGEM MÁQUINA	4
SECÇÃO DO LEITOR	6
PROGRAMAS ZX81	
Caminho das Estrelas	7
13	8
Regime Alimentar	10
Dia da Semana	16
PROGRAMAS ZX SPECTRUM	
Auxiliar de Desenho	17
Space Roller	18
Alunagem	20
PROGRAMAS NEWERAIN	
Programa para computador NewBrain ...	21
INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA ...	22

N O T A D O C L U B E Z - 8 0

Muitos já terão dito "afinal a nova apresentação gráfica do nosso boletim foi sol de pouca dura!".

Pois é... Efectivamente, ainda não é possível suportarmos as despesas que a impressão origina. O número anterior foi uma experiencia que arriscámos tendo sobretudo em vista alargar o CLUBE Z-80 a novos utilizadores que o desconhecem.

Pensou-se, no mês de Março, que o boletim deveria ser enviado a um grande número de pessoas (não sócias mas possuidoras de micro-computadores) que poderiam estar interessadas em associar-se ao CLUBE Z-80.

Ora, dado que teríamos de fotocopiar uma quantidade de boletins muito superior à usual, porque não experimentar a impressão (tiragem mínima de 500 exemplares)?

Pensamos, no entanto, que o impacto produzido pela nova revista não atingiu o que se esperava - apenas conseguimos mais 25 sócios.

Será que não seleccionamos os destinatários mais indicados? Será que o preço de inscrição parece alto? Será que o boletim não é o que se espera?

Quaisquer que sejam as razões, o certo é que o CLUBE Z-80 quer continuar COM A PARTICIPAÇÃO E A COLABORAÇÃO DE TODOS!

P A S S O A P A S S O

O " B - A - B A " D O B A S I C

(continuado do nº anterior)

A QUARTA REGRA É NÃO MISTURAR AS INICIALIZAÇÕES, O PROGRAMA PRINCIPAL E OS SUB-PROGRAMAS, MAS COLOCAR CADA CATEGORIA DE MODO BEM DISTINTO NA SEQUENCIA DAS LINHAS DE INSTRUÇÕES BASIC

A tendência natural é colocar as três categorias na ordem adoptada pela enumeração precedente. Mas é necessário reflectir: O que distingue um sub-programa do programa principal? Em geral, determina-se que uma parte de programa será um sub-programa, devido a uma das seguintes razões:

- trata-se de uma sequencia de operações a ser executadas em muitos locais do programa; portanto, é preferível escrever as instruções uma só vez.

- trata-se de pequenas partes de programas complexos; portanto a realização de sub-programas é uma questão de modularidade.

Relativamente à primeira razão, resulta que os sub-programas serão indubitavelmente as partes de programa executadas mais frequentemente que as outras. Ora, o que faz o BASIC interpretado quando deve ser executada uma instrução do tipo:

GOTO 1000 ou
GOSUB 1000 ?

Ele explora o programa desde o início até localizar a linha respectiva (no exemplo dado é a linha 1000). Portanto, se queremos que essa execução seja rápida, é vantajoso fazer com que as partes mais frequentemente executadas estejam o mais perto possível do início do programa. É por isso que deverá evitar-se inserir os sub-programas depois do fim do programa principal.

Para além disso, se alguns dos sub-programas estabelecidos são muito específicos para a aplicação concernente, outros serão de interesse geral e re-utilizáveis. Se se pretende formar uma biblioteca de sub-programas, será mais cómodo que eles estejam num lugar fixo.

Com um pouco de prática, é fácil lembrar mos o número de linha em que começa certo sub-programa utilizado frequentemente. Enfim, nas aplicações de informática individual, o programa principal começa quase sempre pela exibição de um "Menu" que apresenta o conjunto das grandes funções que o programa oferece ao utilizador. Será mais cómodo que a parte do programa que inclui este menu esteja sempre num lugar fixo: encontrá-la-emos facilmente.

Todas estas considerações levam a adoptar uma regra para a numeração, e a respeitá-la aquando da escrita de todos os programas posteriores. Por exemplo, poderá decidir-se de uma vez para sempre que:

- as linhas 0 a 990 serão reservadas para as inicializações; elas terminarão por um deslocamento do início do programa, de modo que as instruções de inicializações sejam suprimidas após execução.

- as linhas 1000 a 2990 serão reservadas para os sub-programas

- a afixação do menu começará na 3000

- a resposta 1 ao menu enviará à linha 11 000, a resposta 2 à linha 12 000, e as sim sucessivamente.

Naturalmente, coloca-se em 999 um GOTO 3000, de modo a evitar que o programa siga automaticamente para sub-programas.

A adopção de tais regras apresenta muitas vantagens:

- os sub-programas são re-utilizáveis: antes de escrever um novo programa, deve carregar-se o programa precedente para a memória e eliminar-se-á tudo o que precede e tudo o que se segue aos sub-programas, tal como os sub-programas específicos.

- os sub-programas frequentes serão sempre chamados pelo mesmo número de instrução BASIC.

- o conhecimento do menu, facilmente detectável devido à sua posição fixa, permite uma orientação cómoda; a qualquer momento sabemos em que grande capítulo do programa nos encontramos, de acordo com o número de linha da instrução.

- o programa executa-se mais rapidamente

A QUINTA REGRA É NÃO FAVORECER A CONFUSÃO NAS VARIÁVEIS

As variáveis locais que são utilizadas nos sub-programas e os ciclos liberados à saída devem ser claramente distintos das variáveis que suportam resultados intermediários, dados e indicadores, assim como variáveis que servem para transmitir parâmetros. Em caso de ciclos e sub-programas imbricados, isso pode evitar erros. Pode-se, por exemplo, decidir adoptar:

- nomes de variáveis começando de A a H para os dados e os resultados intermediários
 - nomes de variáveis começando de I a P para as variáveis locais de ciclos
 - nomes de variáveis começando de R a Z para as variáveis locais de sub-programas
 - nomes compostos por uma letra seguida de um algarismo para os parâmetros que se transmitem entre programa principal e sub-programas
 - nomes compostos por duas letras para as outras variáveis
- De facto, pouco importam as convenções, desde que se adopte e respeite uma regra.

A SEXTA REGRA CONSISTE EM DOCUMENTAR O PROGRAMA

Quando puser o programa a funcionar pela primeira vez, é possível que ainda se lembre bem das diferentes partes do programa. Mas os problemas tratados pela informática evoluem frequentemente e, ao fim de alguns meses, quando você quiser alterar o seu programa, os comentários que lhe tiver inserido inicialmente são extremamente preciosos. Arranje tempo para os redigir de modo claro, conciso e completo. Termine os blocos de programa por uma linha REM em branco, de modo a que cada grupo lógico se destaque com nitidez.

A SÉTIMA REGRA É RESPEITAR O UTILIZADOR

Esta regra implica que, uma vez o programa terminado, sejam feitos testes para verificar o seu funcionamento.

O utilizador tem direito a errar. Quando isso acontece é talvez depois de ter repetido 100 vezes a mesma operação sem se enganar. Deve-se pensar, a todo o momento, no conforto do utilizador; se for inevitável que ele leia uma mensagem de erro, que ao menos ela seja em português e agradável. Parta do princípio que o utilizador fará cada vez mais erros se começar a entrar em pânico. Todos os erros devem ser previstos. Estas observações são também válidas se o utilizador... é você.

Não ponha questões quando há uma única resposta a desencadear uma acção. Por exemplo: QUER VOLTAR AO MENU (S/N), é desnecessária; ESCREVA "ENTER" PARA VOLTAR AO MENU é suficiente.

Cuide das afixações e das edições de modo a facilitar a sua consulta.

Resista à tentação de usar processos manhosos, mesmo que eles o seduzam no momento.

Economizar um segundo de tratamento ou ganhar três linhas de programa não valem para perder uma estrutura clara. E não esqueça que "menos linhas de Basic" não significa sempre "mais rapidez de execução".

Na informática tradicional, de há cerca de 15 anos, o ponto de vista era totalmente diferente. Os tratamentos faziam-se por lotes e não pelo modo de "diálogo", e por vezes demoravam muitas horas. Actualmente, uma hora de aluguer de um grande computador pode valer 7 vezes mais o salário mensal de um programador. Se se puder ganhar uma hora num tratamento de 5 horas, já é uma economia apreciável.

Dantes, os programas eram estabelecidos com o único objectivo de ser o mais eficientes possível. O melhor programa era o que economizava memória ou se executava mais rapidamente. Hoje, o seu tempo é muito mais precioso que o do computador. Por isso, considera-se um terceiro critério tão importante como os dois primeiros: o programa deve ser fácil de ler e de modificar, lego, estético. De qualquer modo, o autor deve agora ter um bom estilo!

(Traduzido e adaptado da revista "L'ORDINATEUR INDIVIDUEL", Nov.82)

FERNANDO D'ALMEIDA PRECES

CAPÍTULO I

1.1 - O ZX81 possui um interpretador BASIC (memorizado na ROM - 8K bytes de instruções em código máquina) que é a alma da simplicidade do seu manejo, mas que introduz algumas restrições no capítulo respeitante à entrada da programação em código máquina.

Existem vários processos de ultrapassar essa dificuldade, todos passando pela utilização de pequenos programas BASIC que além de serem, na maior parte dos casos, o suporte dessas rotinas, também servem de interligação entre elas, porque todas as vezes que tenhamos de introduzir dados, ou simplesmente intervir no teclado, para a normal sequência dum programa, terá de ser feito através de linhas de instruções em BASIC.

Falemos primeiro do processo mais vulgarizado (as instruções REM na primeira linha da programação BASIC) aonde ficam reservados os caracteres necessários às instruções (USR), à introdução de dados ou ainda de textos, tabelas, etc.

Muitos de nós já sabem por experiência que escrever uma REM muito extensa se torna fastidioso, pois à medida que esta cresce o processo fica mais lento. Como teremos de trabalhar com REMs muito extensas, vamos elaborar um programa monitor cuja primeira rotina tem por missão criar grandes REMs, a partir duma inicial por nós elaborada.

1.2 - Programa Monitor - Rotina 1

Este programa terá mais rotinas para outros fins, mas por agora escreva:

```

20 REM 12345678901234567890
100 PRINT AT 2,6;"PROGRAMA MONI
TOR"
105 PRINT ,,"TAB 11;"INDICE"
110 PRINT ,,"1- RESERVA DE ESPA
CO EM REMS"
200 PRINT AT 20,0;"INDIQUE NUME
RO DA ROTINA"
205 INPUT N
210 CLS
215 GOSUB 10**4-N*100
220 CLS
225 LIST
9900 PRINT AT 2,0;"DIGA QUANTAS
REMS IGUAIS TEM ?"
9902 INPUT N
9904 PRINT ,,"DIGA O NUMERO DE
CARACTERES/REM ?"

```

```

9906 INPUT X
9908 DIM R(N)
9910 LET I=16509
9912 LET T=N*(X+6)+(I-1)
9914 LET B=N*(X+6)-4
9916 IF B<256 THEN GOTO 9980
9918 LET C=INT (B/256)
9920 LET D=((B/256)-C)*256
9922 POKE I+2,D
9924 POKE I+3,C
9926 FOR A=I TO T-1
9928 IF PEEK A=118 THEN POKE A,0
9930 NEXT A
9932 FOR N=I+5 TO A-2
9934 POKE N,27
9936 NEXT N
9938 RETURN
9980 POKE I+2,B
9982 GOTO 9926

```

Grave esta parte do programa antes de continuarmos, pois se houver algo errado pode ter problemas.

2ª ETAPA - ESCREVER A REM

Na linha 20 do nosso programa, temos uma REM com 20 caracteres. Estes devem ser rigorosamente contados para que o programa possa funcionar bem.

Chame a linha 20 com os comandos SHIFT + EDIT, e passe o número de linha para 21, fazendo-o entrar no programa com NEW LINE. Repita esta operação por exemplo 5 vezes, e ficará com 5 REMs de 20 caracteres cada, com os números de linha: 20, 21, 22, 23, 24 e 25.

RUN o programa e responda aos dois pedidos com a informação (6) e (20) respectivamente. Aguarde uns segundos e poderá verificar que as 6 REMs foram transformadas numa única, com o número de linha (20) e com cerca de 6 vezes o tamanho das anteriores.

Note que a rotina está preparada para responder a outros valores introduzidos, e muito rapidamente. Vamos ver porquê.

RESUMO TEÓRICO SOBRE A ACÇÃO DA ROTINA 1

Vamos supor que tem duas pequenas linhas BASIC, como se segue:

10 REM
20 REM

Estas linhas ficam armazenadas na memória com os seguintes códigos:

C, 10, 2, 0, 234, 118

C, 20, 2, 0, 234, 118

O primeiro byte fica na localização 16509 e o último na 16520.

Os primeiros 2 bytes duma linha BASIC determinam o número de linha, que como todos sabemos pode ir de (1 a 9.999). A distribuição do número decimal, pelos 2 bytes citados, é encontrada pelas seguintes expressões matemáticas:

$INT(N/256) = y$ (1º byte)

$INT(N-Y*256) = x$ (2º byte)

ou a inversa

$y*256+x = N$ (número)

Os dois bytes seguintes representam o número de caracteres dessa linha, sendo o primeiro o menos significativo (baixo byte). A distribuição do número pelos 2 bytes rege-se pelas expressões matemáticas já apresentadas. O byte seguinte é a instrução do sistema BASIC (no exemplo apresentado é a instrução REM). A fechar a linha encontramos a instrução NEW LINE - código 118.

Se nós quisermos transformar estas duas linhas (a 10 e a 20) numa única, teremos de introduzir na localização 16511, o código 8 (a nova linha passa a ter uma extensão de 8 caracteres), e eliminar a instrução NEW LINE em 16514, introduzindo o código de outro qualquer caracter. A nossa rotina 1 executa precisamente este trabalho para o número de REMs e caracteres solicitado.

1.3 - Programa monitor (Introdução de dados ou mensagens na REM) -
- Rotina 2

Uma vez "fabricada" a grande REM com a ajuda da ROTINA 1, temos agora aproximadamente 120 bytes à nossa disposição, para lá introduzir o que quisermos.

Para o nosso primeiro ensaio, vamos lá colocar uma pequena mensagem, com a finalidade de a reproduzir no ecrã numa dada sequência de programação. A introdução dessa mensagem na REM é feita através da seguinte rotina:

```
115 PRINT **,"2- METER DADOS NA  
REM"  
9800 PRINT **,"INTRODUCAO DE DADO  
S"  
9801 PRINT **,"NAO ULTRAPASSE A E  
XTENSAO DA REM"  
9802 LET X=16514  
9803 PRINT **,"PARA FIM DA MENSAG  
EM,PRIMA <*> (CODIGO 216)"  
9804 INPUT A$  
9805 FOR N=1 TO LEN A$  
9806 POKE X,CODE A$(N)  
9810 IF N=LEN A$ THEN GOTO 9820  
9812 PRINT CHR$ CODE A$(N);  
9814 LET X=X+1  
9816 NEXT N  
9818 GOTO 9804  
9820 IF CODE A$(N)=216 THEN RETU  
RN  
9821 PRINT CHR$ CODE A$(N)  
9822 GOTO 9814
```

Faça rolar o programa e seleccione a ROTINA 2. Como primeiro ensaio vamos introduzir na REM uma pequena mensagem, cujo arranjo em grupos de 32 caracteres terá de ser ensaiado para não haver separação imprópria de palavras.

"PROGRAMA 1 - ESTA MENSAGEM SERA INTEGRALMENTE COPIADA PARA O ECRAN, POR ACÇÃO DUMA ROTINA EM BASIC.

PROGRAMA 2 - A MESMA ACÇÃO COM UMA ROTINA EM CODIGO MAQUINA.

PROGRAMA 3 - TRANSMISSAO INSTANTANEA DE UM BLOCO DE MEMÓRIA POR UMA INSTRUÇÃO LDIR.**"

Depois de ter passado este texto para a REM escreva a seguir:

```
1000 REM ROTINA COPIA REM  
1010 LET HL=16514  
1015 LET A=PEEK HL  
1020 LET CP=A-216  
1025 IF NOT CP THEN GOTO 1050  
1030 PRINT CHR$ A;  
1035 LET HL=HL+1  
1040 GOTO 1015  
1050 PRINT **,"FIM DA MENSAGEM"  
1055 STOP
```

NOTA: Os utilizadores de máquinas sem extensão de memória terão de começar por apagar parte do programa, para poderem continuar a experiencia. Posso sugerir que apaguem as ROTINAS 1 e 2. Façam do vosso gravador a extensão que lhes falta.

(continua no próx. nº)

SEÇÃO DO LEITOR

...SUGESTÕES...COMENTÁRIOS...OPINIÕES...DÚVIDAS...SUGESTÕES...COMENTÁRIOS...OPINIÕES...DÚVIDAS

• 1-(...)A revista MICRO SYSTEMES nº28 publicou um artigo sobre a memória RAM de base (1K) do ZX81, a qual poderá ser de 2K, fazendo apenas a seguinte modificação: retirar os dois integrados "2114" e substituí-los por um: o "4802" - mudança essa que poderá ser feita, pois já foi prevista pelo fabricante e o circuito impresso assim o permite. Onde será possível adquirir este circuito integrado, pois os estabelecimentos de venda de componentes electrónicos (no Porto) não o possuem?

2-Esta montagem depois de realizada e com o módulo de 16K ligado, passa a ser de 17K a memória disponível no total ou mantém-se em 16K?

3-Haverá a possibilidade do ZX81 funcionar a cores? (ex: mudar o modulador)

4-Se fizermos, depois de introduzirmos no ZX81 um programa qualquer:

9000 SAVE "ABCDE"

9010 RUN

e depois:

GOTO 9000 (+ NewLine)

é gravado o programa em cassette e feito RUN automaticamente. Em seguida constata-se que a última letra do nome do programa na linha 9000 passa a ser inverso (neste caso a inverso E). A pergunta é: porque acontece isto?"

EDUARDO JORGE / Porto

• O CLUBE Z-80 responde:

1-A substituição pode ser feita com a memória RAM 4118 (1Kx8bit), que o importador da SINCLAIR possui normalmente.

2-Quando o módulo de expansão da memória está ligado, deixa de funcionar a memória interna de 1K (2K no TM1000), e portanto ficará com 16K para utilizar.

3-Temos conhecimento de que em Inglaterra existe um Inteface para TV cor mas pensamos que é pouco mais que a simples mudança do modulador.

4-Os autores do projecto da ROM do ZX81 entenderam que deveria existir um sinal representativo de que a gravação tinha terminado. Desse modo, arbitraram que quando a gravação fosse efectuada, o último carácter do nome do programa ficasse em Inverso Video.

• Mais 4 LIVROS à disposição do CLUBE Z-80

-"COMPUTER PUZZLES: FOR SPECTRUM & ZX81", Ian Stewart & Robin Jones 150\$00

-"INTRODUCING SPECTRUM MACHINE CODE" 300\$00

-"ILLUSTRATING BASIC (A SIMPLE PROGRAMMING LANGUAGE)", Donald Alcock 300\$00

-"30 HOUR BASIC - ZX81", Clive Prigmore 450\$00

(Estes livros podem ser enviados à cobrança, em fotocópia, pelo preço indicado)

• JOSÉ LOBO (Porto) está interessado na compra de um ZX81 usado

Telef: 488946

• CARLOS MOITA (Porto) vende um ZX81

Telef: 493762 e 400584

Por razões de vária ordem, entre as quais o atraso na saída do boletim nº6 que se repercutiu na execução deste, não nos foi ainda possível responder a várias cartas recebidas. Aos que esperam respostas, as nossas desculpas! Oportunamente escreveremos, publicando as respostas no próximo número.


```

0000 REM PROGRAMA TRAD.E MODIF.
0010 POR ALMEIDA PRECES, EM 10/5/82
0020 REM "2"
0030 GOSUB 9400
0040 GOSUB 9000
0050 GOSUB 8000
0060 GOSUB 6950
0070 GOSUB 7000
0080 GOSUB 6950
0090 GOSUB 7500
0100 PRINT "QUAIS AS SUAS ORDE
NS?"
0110 PRINT TAB 5;"SIR?",TAB 12;
"1 - INVESTIGACAO";TAB 12;"2 - M
QUIMENTO";TAB 12;"3 - COMBATE"
0120 INPUT D
0130 IF D<1 OR D>3 THEN GOTO 140
0140 GOSUB 6950
0150 GOSUB 1000*D
0160 FOR W=1 TO 30
0170 PRINT AT 20,5;"|||||";AT 20
,5;"
0180 NEXT W
0190 GOTO 30
0200 PRINT TAB 4;"PESQUISA"
0210 GOSUB 7500
0220 PRINT "CURTO (1) OU", " LONG
O ALCANCE (2) SIR?"
0230 INPUT K
0240 LET E=E-10*K
0250 GOSUB 6950
0260 IF K=2 THEN GOTO 1500
0270 IF A(B+1,C)=1 OR A(B+1,C+1)
=1 OR A(B,C+1)=1 OR A(B-1,C)=1 O
R A(B-1,C-1)=1 OR A(B,C-1)=1 OR
A(B+1,C-1)=1 OR A(B-1,C+1)=1 THE
N PRINT Z$;"... ESTAMOS PROXIMOS
", "SIR"
0280 FOR T=1 TO 150
0290 NEXT T
0300 RETURN
0310 GOSUB 7500
0320 PRINT AT 15,0;"DIRECCAO: N-
1, S-2, E-3, W-4?"
0330 PRINT TAB 8;"(INTRODUZA O N
UMERO)"
0340 INPUT N
0350 LET Z=0
0360 IF N=1 AND A(B-2,C)=1 THEN
LET Z=1
0370 IF N=2 AND A(B+2,C)=1 THEN
LET Z=1
0380 IF N=3 AND A(B,C+2)=1 THEN
LET Z=1
0390 IF N=4 AND A(B,C-2)=1 THEN
LET Z=1
0400 GOSUB 7500
0410 PRINT "INFORM. PESQ. L.ALCA
NCE"
0420 IF Z=1 THEN PRINT "POSITIVA"
0430 IF Z=0 THEN PRINT "NEGATIVA"
0440 FOR T=1 TO 150
0450 NEXT T
0460 RETURN
0470 LET E=E-50
0480 LET A(B,C)=0
0490 LET B(B,C)=0
0500 PRINT "DIRECCAO (N/S)?"
0510 INPUT A$
0520 LET B=B-1
0530 IF A$="S" THEN LET B=B+2
0540 GOSUB 7500
0550 PRINT "NOVA COORDENADA ";B;
"C"
0560 PRINT TAB 12;"(E/W)?"
0570 INPUT A$
0580 LET C=C-1
0590 IF A$="E" THEN LET C=C+2
0600 PRINT "NOVA COORDENADA ";B;
"C"
0610 IF A(B,C)=1 THEN GOTO 5500
0620 LET A(B,C)=2
0630 LET B(B,C)=2
0640 GOSUB 8000
0650 RETURN
0660 GOSUB 7500
0670 PRINT "DIRECCAO DO LAISER (
N/S)?"
0680 INPUT A$
0690 LET G=B-1
0700 IF A$="S" THEN LET G=G+2
0710 PRINT TAB 12;"(E/W)?"

```

```

3120 LET F=C-1
3130 INPUT A$
3140 IF A$="E" THEN LET F=F+2
3150 LET E=E-100
3160 IF A(G,F)<>1 THEN GOTO 3300
3170 GOSUB 7500
3180 PRINT "FOI ATINGIDO O ";Z$
3190 LET AL=AL+1
3200 LET B(G,F)=4
3210 RETURN
3220 GOSUB 7500
3230 LET B(G,F)=3
3240 PRINT "ATENCAO, MISSIEIS...
SOMRE NOS."
3250 FOR G=1 TO 30
3260 NEXT G
3270 PRINT "O ";Z$;"FOI TOCADO"
3280 FOR G=1 TO 30
3290 NEXT G
3300 GOSUB 6950
3310 IF RND>.6 THEN GOTO 3400
3320 PRINT "ELES TOCARAM-NOS, SI
R"
3330 GOSUB 8000
3340 LET E=E-100*RND
3350 RETURN
3360 PRINT "O ";Z$;" ATENGIDO, S
IR"
3370 RETURN
3380 GOSUB 6950
3390 SCROLL
3400 PRINT "BANCO DE ENERGIA ESG
OTADO"
3410 SCROLL
3420 PRINT "FORAM ABATIDAS ";AL;
" NAVES"
3430 SCROLL
3440 PRINT "NESTA MISSAO"
3450 SCROLL
3460 PRINT "O SEU COMANDO VALEU
";INT (AL/5*100);" PONTOS."
3470 GOTO 3805
3480 GOSUB 6950
3490 FOR T=1 TO 100
3500 PRINT AT 15,0;"A SUA NAVE E
ABATEU NA NAVE ";Z$
3510 PRINT AT 15,0;"A SUA NAVE E
ABATEU NA NAVE "
3520 PRINT AT 17,3;"E N E R G E
N C I A"
3530 PRINT AT 17,3;"E N E R G E
N C I A"
3540 NEXT T
3550 PRINT "505...COMBATE CORPO
A CORPO...505...505...";
3560 STOP
3570 PRINT AT 13,0;"

```

```

6955 PRINT AT 13,0;
6970 RETURN
7020 PRINT AT 2,14;"ENERGY BANK:
";INT E;"
7030 IF E<1 THEN GOTO 3800
7040 IF AL>0 THEN PRINT AT 3,14;
"NAVES ABATIDAS";AT 4,17;"NUMERO
";AL
7050 PRINT AT 7,14;"VOCE ESTA EM
";B;" ";C
7070 PRINT AT 8,14;"
7075 PRINT AT 8,14;"NÃO ";
7080 GOSUB 8500
7100 PRINT "SECTOR"
7120 PRINT AT 12,0;
7130 RETURN
7500 LET R=INT (RND*5)
7520 IF R=0 THEN PRINT "SPOCK:"
7540 IF R=1 THEN PRINT "SCOTT:"
7560 IF R=2 THEN PRINT "LT. UHUR
A:"
7580 IF R=3 THEN PRINT "CHECKOV:"
7600 IF R=4 THEN PRINT "SULU:"
7620 RETURN
8005 PRINT AT 0,0;
8010 PRINT "1234567890"
8020 FOR Q=1 TO 10
8025 PRINT AT 0,13;" ";AT 0,13;"

```



```

8030 IF Q<10 THEN PRINT Q;" ";
8035 IF Q=10 THEN PRINT Q;
8040 FOR P=1 TO 10
8050 IF B(Q,P)=0 THEN PRINT " ";
8060 IF B(Q,P)=2 THEN PRINT " ";
8100 IF B(Q,P)=3 THEN PRINT " ";
8120 IF B(Q,P)=4 THEN PRINT " ";
8160 NEXT P
8200 NEXT Q
8210 PRINT
8220 PRINT " 1234567890"
8490 RETURN
8520 LET Q=B*C
8540 IF Q<10 THEN PRINT "ANTARES"
8560 IF Q>9 AND Q<20 THEN PRINT "RIGEL";
8580 IF Q>19 AND Q<30 THEN PRINT "PROCYON";
8600 IF Q>29 AND Q<40 THEN PRINT "VEGA";
8620 IF Q>39 AND Q<50 THEN PRINT "CANOPUS";
8640 IF Q>49 AND Q<60 THEN PRINT "ALTAIR";
8660 IF Q>59 AND Q<70 THEN PRINT "SAGITTARIUS";
8680 IF Q>69 AND Q<80 THEN PRINT "POLLUX";
8700 IF Q>79 AND Q<90 THEN PRINT "SIRIUS";
8720 IF Q>89 THEN PRINT "BETELGEUSE";
8740 RETURN
8999 STOP
9000 FAST
9010 DIM A(10,10)
9020 DIM B(10,10)
9060 FOR A=1 TO 20
9080 LET X=INT (RND*10+1)
9100 LET Y=INT (RND*10+1)
9120 LET A(X,Y)=1
9140 NEXT A
9160 LET B=5
9180 LET C=5
9200 LET A(B,C)=2
9220 LET B(B,C)=2
9240 LET AL=0
9260 LET E=RND
9280 IF E<.33 THEN LET Z$="BRARKONS"
9300 IF E>.33 AND E<.66 THEN LET Z$="WRERKTONIONS"
9320 IF E>.66 THEN LET Z$="POLLUXIANS"
9340 LET E=1000+2000*RND
9345 SLOW
9349 RETURN
9400 PRINT "PATRULHA ESPACIAL"
9410 PRINT "A SUA NAVE TEM DE PATRULHAR UMA"
9420 PRINT "ZONA DO ESPACO, MUITO PERIGOSA."
9430 PRINT "VOCE SABE QUE ESTA RODEADO POR"
9440 PRINT "INIMIGOS QUE SAO INVISIVEIS NOS"
9450 PRINT "SEUS ECRANS E SOMENTE O SEU PES-"
9460 PRINT "QUISADOR OS DETECTA, MAS NAO LHE"
9470 PRINT "DA DIRECCAO DEFINIDA LIMPE ESSA"
9480 PRINT "AREA E OBTENHA A SUA PONTUACAO."
9490 PRINT "O SINAL (+) INDICA A POSICAO DA"
9500 PRINT "NAVE INIMIGA ABATIDA E O (*) IN-"
9510 PRINT "DICA A ANTERIOR POSICAO DE OUTRA"
9520 PRINT "OU SEJA, CAMINHO LIVRE."
9530 PRINT "NOUTROS CASOS UTILIZAMOS SEMPRE OS"
9540 PRINT "PESQUISADORES ANTES DE SE MOVER."
9550 PRINT "PRIMA **N/L** PARA COMECAR."
9560 INPUT L$
9570 CLS
9580 RETURN

```

PROGRAMA "13" - Um jogo para o ZX81
Autor: FERNANDO PRECES

```

0000 REM PROGRAMA TRAD. E MODIF. POR ALMEIDA PRECES EM 21/5/82
2 REM "16"
5 GOSUB 1000
100 LET A$=""
105 LET N=0
110 LET A=0
115 LET B=0
120 LET C=0
125 LET D=0
130 LET T=1
135 LET X=12
160 LET R=INT (RND*30)
170 PRINT AT 21,R;A$
180 SCROLL
190 SCROLL
200 LET N=N+T
205 IF N=100 THEN LET A$=" "
215 LET E=D
220 LET D=C
230 LET C=B
240 LET B=A
250 LET A=R
255 PRINT AT 9,X-2;" "
260 PRINT AT 11,X;" "
270 IF X>=E-T AND X<=E+T THEN GOTO 500
280 IF INKEY$="Z" THEN LET X=X-T-1
285 IF X<=-2 THEN LET X=2
290 IF INKEY$="M" THEN LET X=X+T+1
295 IF X>=30 THEN LET X=30
300 GOTO 160
500 FOR A=1 TO 5
502 PRINT AT 11,X-1;" "
505 PRINT AT 11,X-1;" "
508 PRINT AT 11,X-1;"CRASH"
507 PRINT AT 11,X-1;" "
508 NEXT A
510 PRINT AT 0,0;"SCORE=";N
600 FOR D=1 TO 200
605 NEXT D
610 CLS
620 RUN 100
1000 PRINT AT 3,0;"A NUvem DE AS TERQIDES E ATAQUE"
1005 PRINT "SURPRESA DO INIMIGO"
1010 PRINT "A TECLA ""Z"" MOVE A NAve PARA A"
1015 PRINT "ESQUERDA E A ""M"" MOVE A NAve PARA"
1016 PRINT "A DIREITA."
1020 PRINT "PRIMA **N/L** PARA COMECAR"
1025 INPUT H$
1030 CLS
1035 RETURN

```

• Sobre o programa "REGIME ALIMENTAR", J.F.G.
(Página 10)

Para o ZX81 (com ext.memória 32K-octetos)

"Um equilibrado e saudável regime alimentar deve obedecer a certas normas cientificamente estabelecidas. Acontece, porém, que os calculos a efectuar sem serem complicados, são extensos e morosos, além de exigirem a consulta de tabelas apropriadas. Daí que os nutricionistas em geral se limitem a regimes típicos, mais ou menos estandarizados e simplificados.

A facilidade com que o computador armazena dados fornecidos e a rapidez com que efectua as operações mais intrincadas habilita-o a estabelecer com rigor uma ementa apropriada a cada pes-

soa e a cada caso. É essa finalidade do presente programa. Em função do sexo e da idade (especificada até em meses, como será útil sobretudo para crianças pequenas), ele começa por calcular a altura média normal para Portugal; em seguida, entrando-se com a altura real, calcula o peso ideal; depois, entrando-se com determinadas variáveis (intensidade de trabalho, tempo de descanso, temperatura média do ambiente), calcula a energia dispendida diariamente.

A partir dessa energia é que se vai estabelecer o regime - destinado a fornecer ao organismo as calorias gastas, e isso não apenas na devida quantidade (a fim de não engordar nem emagrecer indevidamente), mas ainda nas correctas proporções dos principais nutrientes: glícidos, prótidos e gorduras - proporções que, a não serem observadas, podem originar várias doenças.

Quem ignore o sentido destes termos deve pelo menos ficar a saber que glícidos (ou hidratos de carbono) são os nutrientes mais típicos dos alimentos farináceos como arroz, batatas, massas pão; e prótidos (ou proteínas) são os nutrientes específicos dos alimentos de origem animal, como carnes, peixes, ovos, queijos, e também abundantes nas sementes de plantas leguminosas, como feijão, ervilha, fava, soja

O programa apresenta cerca de 60 variedades de alimentos, conglomerados nos seguintes grupos:

- Grupo L - Leite e derivados
- Grupo C - Carnes e ovos
- Grupo P - Pescados (peixes e mariscos)
- Grupo A - Produtos amiláceos
- Grupo H - Produtos hortícolas
- Grupo F - Frutas
- Grupo D - Doces
- Grupo G - Gorduras (lípidos)

Cada grupo aparece no ecrã separadamente; o primeiro automaticamente, os outros, à medida que forem sendo chamadas pela letra respectiva.

Cada variedade de alimento, ou de pequenos conjuntos de alimentos idênticos, vem precedida de um número: esse número é que vai servir para introduzir no computador o alimento pretendido (é mais fácil escrever um número de um ou dois algarismos do que um nome).

A quantidade de alimento introduz-se em gramas (que para líquidos corresponde mais ou menos a mililitros ou centímetros cúbicos).

Um exemplo: Suponhamos que se pretende utilizar por dia 150 gramas de pão.

O pão aparece no grupo A (dos amiláceos), e tem o número 25: entra-se com os números 25 (número de referência do pão) e 150 (os 150 gramas de pão que estabelecemos) - devendo estes dois números ser separados por um ou mais espaços - mas fazendo uma só entrada para os dois. Neste caso seria: 25 150 (e Newline).

De notar, mais uma vez, que as quantidades a introduzir devem ser em princípio quantidades diárias. Quem use determinado alimento apenas uma ou duas vezes por semana, deverá previamente calcular a quanto isso corresponde por dia...

Certas pessoas poderão sentir alguma dificuldade em avaliar o peso dos alimentos. Para lhes facilitar a tarefa, vão aqui algumas referências:

Uma colher de sopa bem cheia (acogulada) de arroz cru, feijão cru, açúcar e coisas semelhantes, pesa 20 gramas; um pão, 50 gramas; um ovo, uns 60 gr; uma peça de fruta média, de 100 a 150 gr; uma batata média, uns 100gr; um bife ou uma posta de peixe, 150 a 200 gr; uma chavena almoçadeira de leite, uns 200 gramas ...

Como não é de esperar que logo à primeira tentativa se obtenha a ementa equilibrada, é conveniente ir tomando nota das entradas que se vão fazendo. Assim torna-se mais fácil corrigir excessos ou defeitos: uma nova entrada de qualquer alimento anula a entrada anterior do mesmo alimento. De qualquer grupo se pode tomar uma ou mais variedades de alimentos (ou nenhuma ...); e em qualquer altura se pode passar a outro grupo, bastando premir a letra respectiva (e NL); e também em qualquer altura se pode saber como vai a soma dos nutrientes, premindo-se S (e NL). Como curiosidade, o programa também indica o preço total dos respectivos produtos no mercado (preços correntes em princípio de 1983). Os valores finais são arredondados a +5 (ou +10), mas os valores parciais com que o computador opera, esses são o mais exactos possível. Uma vez estabelecida a ementa conveniente, para recomençar, destruindo as variáveis introduzidas, premir R (e NL).

Antes de algumas entradas, enquanto no ecrã se projectam instruções, há PAUSAS, que se podem abreviar premindo qualquer letra.

Bom regime alimentar, boa saúde e boa sorte!" J.F.G. / Braga

ZX 81PROGRAMA "REGIME ALIMENTAR"

Autor: J.F.G. / Braga

```

10 PRINT "....."PARA GOZAR DE BOA SAUDE A PARTIR"
11 PRINT "DUM EQUILIBRADO REGIME ALIMENTAR"
12 PRINT "  RESPONDA A TODAS AS PERGUNTAS"
13 PRINT "  QUE O COMPUTADOR LHE FORMULAR"
15 PAUSE 500
16 POKE 16437,255
18 CLS
21 PRINT "O SEU SEXO?      (M OU F)"
25 INPUT S$
26 CLS
27 PRINT "ANOS COMPLETOS DE IDADE?"
30 INPUT IA
31 CLS
32 PRINT "ALEM DOS ";IA;" ANOS, QUANTOS MESES?"
40 INPUT IM
41 CLS
42 LET I=IA+IM/12
43 IF I<=20 THEN GOTO 47
44 IF S$="M" THEN LET AI=175
45 IF S$="F" THEN LET AI=162
46 GOTO 50
47 IF S$="M" THEN LET AI=INT (SIN ((I*1.1+20)/30)*263-88)+.5
48 IF S$="F" THEN LET AI=INT (SIN ((I*1.1+20)/30)*229-67)+.5
51 PRINT "V. DEVERIA MEDIR ";AI;" CENTIMETROS"
56 PRINT "QUANTO MEDE?      (EM CENTIMETROS)"
59 INPUT A
60 CLS
61 IF I<=20 THEN GOTO 70
62 LET I1=I
63 IF I>50 THEN LET I1=50
66 IF S$="M" THEN LET P=INT (((A-175)*A/225+(I1-20)*.2+65)*10+.5)/10
67 IF S$="F" THEN LET P=INT (((A-162)*A/240+(I1-20)*.18+55)*10+.5)/10
68 GOTO 72
70 IF S$="M" THEN LET P=INT ((TAN ((A+109)/210)*16.52-9.4)*10+.5)/10
71 IF S$="F" THEN LET P=INT ((TAN ((A+82)/180)*13.35-6)*10+.5)/10
75 PRINT "O SEU PESO ""IDEAL"": ";P;" KG"
83 PAUSE 300
84 POKE 16437,255
86 CLS
100 IF I<=12 THEN LET K=INT ((P*(112-3.75*I))+.5)
102 IF I>12 AND I<=25 AND S$="M" THEN LET K=P*(320/I**((1/1.39)+13.5))
104 IF I>12 AND I<=25 AND S$="F" THEN LET K=P*(605/I**((1/1.13)))
106 IF I>25 AND S$="M" THEN LET K=P*(100/SQR I+25)
108 IF I>25 AND S$="F" THEN LET K=P*(80/SQR I+19)
135 PRINT "VALORIZE, NUMA ESCALA DE 0 A 10,"
136 PRINT "A INTENSIDADE DO TRABALHO DIARIO"
150 INPUT T
151 CLS
155 LET K=K*(1.1**((T-5)))

```



```

160 PRINT "QUANTAS HORAS DE SONO?"
170 INPUT R
171 CLS
172 LET K=K+50*((24-R)-16)
173 PRINT "TEMPERATURA DO AMBIENTE?"
175 INPUT TA
176 CLS
177 LET K=INT ((K+(15-TA)*20)/10+.5)*10
181 PRINT ,,,,"V. ""GASTA"" DIARIAMENTE CERCA DE"
182 PRINT K;" CALORIAS BIOLOGICAS (KCAL). "
183 PAUSE 400
184 POKE 16437,255
185 CLS
187 PRINT "VAO APARECER GRUPOS DE ALIMENTOS"
188 PRINT ,,"UMA LETRA IDENTIFICA CADA GRUPO;"
189 PRINT "UM NUMERO IDENTIFICA O ALIMENTO."
190 PRINT ,,,,"PREMINDO A LETRA APARECE O GRUPO"
191 PRINT "(PREMINDO S, FAZEM-SE AS SOMAS,"
192 PRINT "PREMINDO R, VAI-SE AO PRINCIPIO)"
193 PRINT ,,,,"INTRODUZ-SE O NUMERO DO ALIMENTO"
194 PRINT "SEGUIDO DE UM INTERVALO OU DOIS"
195 PRINT "E DA QUANTIDADE DIARIA EM GRAMAS"
202 DIM A(59,4)
204 DIM B(59,4)
206 DIM E(59)
211 LET A(1,1)=3.3
212 LET A(1,2)=3.5
213 LET A(1,3)=4.8
214 LET A(1,4)=20
221 LET A(2,1)=3.6
222 LET A(2,2)=.5
223 LET A(2,3)=4.9
224 LET A(2,4)=20
231 LET A(3,1)=9
232 LET A(3,2)=28
233 LET A(3,3)=3.2
234 LET A(3,4)=250
241 LET A(4,1)=26
242 LET A(4,2)=25
243 LET A(4,3)=2
244 LET A(4,4)=380
251 LET A(5,1)=25
252 LET A(5,2)=30
253 LET A(5,3)=2
254 LET A(5,4)=520
261 LET A(6,1)=20
262 LET A(6,2)=2
263 LET A(6,3)=1
264 LET A(6,4)=400
271 LET A(7,1)=3.5
272 LET A(7,2)=3
273 LET A(7,3)=4.5
274 LET A(7,4)=250
281 LET A(8,1)=.4
282 LET A(8,2)=85
283 LET A(8,3)=.5
284 LET A(8,4)=400
291 LET A(9,1)=12
292 LET A(9,2)=12
293 LET A(9,3)=.5
294 LET A(9,4)=120
301 LET A(10,1)=20
302 LET A(10,2)=10
303 LET A(10,3)=.2
304 LET A(10,4)=400
311 LET A(11,1)=20
312 LET A(11,2)=5
313 LET A(11,3)=.2
314 LET A(11,4)=150
321 LET A(12,1)=17
322 LET A(12,2)=25
323 LET A(12,3)=.2
324 LET A(12,4)=350
331 LET A(13,1)=20
332 LET A(13,2)=4
333 LET A(13,3)=.2
334 LET A(13,4)=280
341 LET A(14,1)=17
342 LET A(14,2)=12
343 LET A(14,3)=.2
344 LET A(14,4)=250
351 LET A(15,1)=22
352 LET A(15,2)=6
353 LET A(15,3)=.5
354 LET A(15,4)=250
361 LET A(16,1)=9
362 LET A(16,2)=40
363 LET A(16,3)=.2
364 LET A(16,4)=400
371 LET A(17,1)=20
372 LET A(17,2)=12
373 LET A(17,3)=.1
374 LET A(17,4)=350
381 LET A(18,1)=18
382 LET A(18,2)=6
383 LET A(18,3)=.1
384 LET A(18,4)=300
391 LET A(19,1)=17
392 LET A(19,2)=1
393 LET A(19,3)=.1
394 LET A(19,4)=500
401 LET A(20,1)=40
402 LET A(20,2)=1

```



```

403 LET A(20,3)=.1
404 LET A(20,4)=400
411 LET A(21,1)=18
412 LET A(21,2)=1.5
413 LET A(21,3)=.3
414 LET A(21,4)=350
421 LET A(22,1)=25
422 LET A(22,2)=20
423 LET A(22,3)=.1
424 LET A(22,4)=400
431 LET A(23,1)=10
432 LET A(23,2)=1
433 LET A(23,3)=.1
434 LET A(23,4)=250
441 LET A(24,1)=18
443 LET A(24,2)=2
443 LET A(24,3)=.1
444 LET A(24,4)=1500
451 LET A(25,1)=8.5
452 LET A(25,2)=1.5
453 LET A(25,3)=55
454 LET A(25,4)=100
461 LET A(26,1)=6.5
462 LET A(26,2)=1
463 LET A(26,3)=80
464 LET A(26,4)=45
471 LET A(27,1)=6.7
472 LET A(27,2)=1
473 LET A(27,3)=80
474 LET A(27,4)=50
481 LET A(28,1)=2.2
482 LET A(28,2)=.1
483 LET A(28,3)=18
484 LET A(28,4)=25
491 LET A(29,1)=22
492 LET A(29,2)=1.5
493 LET A(29,3)=58
494 LET A(29,4)=90
501 LET A(30,1)=40
502 LET A(30,2)=3
503 LET A(30,3)=30
504 LET A(30,4)=120
511 LET A(31,1)=24
512 LET A(31,2)=.1
513 LET A(31,3)=53
514 LET A(31,4)=75
521 LET A(32,1)=27
522 LET A(32,2)=47
523 LET A(32,3)=15
524 LET A(32,4)=150
531 LET A(33,1)=3
532 LET A(33,2)=.2
533 LET A(33,3)=3
534 LET A(33,4)=60
541 LET A(34,1)=2
542 LET A(34,2)=.3
543 LET A(34,3)=6
544 LET A(34,4)=80
551 LET A(35,1)=4
552 LET A(35,2)=.1
553 LET A(35,3)=10
554 LET A(35,4)=90
561 LET A(36,1)=.7
562 LET A(36,2)=.1
563 LET A(36,3)=6
564 LET A(36,4)=35
571 LET A(37,1)=2
572 LET A(37,2)=.2
573 LET A(37,3)=8
574 LET A(37,4)=50
581 LET A(38,1)=3
582 LET A(38,2)=.2
583 LET A(38,3)=12
584 LET A(38,4)=40
591 LET A(39,1)=1
592 LET A(39,2)=.4
593 LET A(39,3)=3.5
594 LET A(39,4)=50
601 LET A(40,1)=1
602 LET A(40,2)=.2
603 LET A(40,3)=7
604 LET A(40,4)=80
611 LET A(41,1)=.35
612 LET A(41,2)=.45
613 LET A(41,3)=14
614 LET A(41,4)=50
621 LET A(42,1)=.8
622 LET A(42,2)=.5
623 LET A(42,3)=12
624 LET A(42,4)=75
631 LET A(43,1)=.75
632 LET A(43,2)=.2
633 LET A(43,3)=10
634 LET A(43,4)=100
641 LET A(44,1)=.5
642 LET A(44,2)=1
643 LET A(44,3)=18
644 LET A(44,4)=70
651 LET A(45,1)=1
652 LET A(45,2)=1
653 LET A(45,3)=10
654 LET A(45,4)=120
661 LET A(46,1)=1.5
662 LET A(46,2)=.5
663 LET A(46,3)=22
664 LET A(46,4)=130
671 LET A(47,1)=4
672 LET A(47,2)=3
673 LET A(47,3)=40
674 LET A(47,4)=90
681 LET A(48,1)=15
682 LET A(48,2)=50
683 LET A(48,3)=20
684 LET A(48,4)=150
691 LET A(49,1)=.4
692 LET A(49,2)=.1
693 LET A(49,3)=99.5
694 LET A(49,4)=60
701 LET A(50,1)=.5
702 LET A(50,2)=.2
703 LET A(50,3)=78
704 LET A(50,4)=500
711 LET A(51,1)=3
712 LET A(51,2)=24
713 LET A(51,3)=70
714 LET A(51,4)=1000
721 LET A(52,1)=3
722 LET A(52,2)=.5
723 LET A(52,3)=60
724 LET A(52,4)=400
731 LET A(53,1)=12
732 LET A(53,2)=6
733 LET A(53,3)=80
734 LET A(53,4)=150
741 LET A(54,1)=10
742 LET A(54,2)=1.5
743 LET A(54,3)=75
744 LET A(54,4)=350
751 LET A(55,1)=.2
752 LET A(55,2)=.5
753 LET A(55,3)=65
754 LET A(55,4)=400
761 LET A(56,1)=1
762 LET A(56,2)=1
763 LET A(56,3)=70
764 LET A(56,4)=300
771 LET A(57,1)=.01
772 LET A(57,2)=100
773 LET A(57,3)=.01
774 LET A(57,4)=300
781 LET A(58,1)=.01
782 LET A(58,2)=100
783 LET A(58,3)=.01
784 LET A(58,4)=180
791 LET A(59,1)=.01
792 LET A(59,2)=85
793 LET A(59,3)=.01
794 LET A(59,4)=200
795 SLOW
850 PAUSE 1000
860 POKE 16437,255
876 CLS

```



```

880 PRINT "GRUPO L - LEITE E DERIVADOS"
881 PRINT "1 - LEITE DE VACA (INTEIRO)"
882 PRINT "2 - LEITE DE VACA DESNATADO"
883 PRINT "3 - QUEIJO CASEIRO"
884 PRINT "4 - QUEIJO TIPO FLAMENGO"
885 PRINT "5 - QUEIJO GORDO"
886 PRINT "6 - QUEIJO MAGRO"
887 PRINT "7 - IOGURTE"
888 PRINT "8 - MANTEIGA"
889 PRINT "GRUPO SEGUINTE: C"
890 GOTO 1000
891 CLS
892 PRINT "GRUPO C - CARNES E OVOS"
893 PRINT "9 - OVOS (CADA OVO, 50 A 60 G)"
894 PRINT "10 - VACA"
895 PRINT "11 - FRANGO"
896 PRINT "12 - PORCO"
897 PRINT "13 - COELHO"
898 PRINT "14 - CARNEIRO"
899 PRINT "15 - FIGADO"
900 PRINT "16 - SALSICHARIA"
901 PRINT "GRUPO SEGUINTE: P"
902 GOTO 1000
903 CLS
904 PRINT "GRUPO P - PESCADOS"
905 PRINT "17 - PEIXE GORDO (ATUM, ENGUIA)"
906 PRINT "18 - PEIXE MEIO GORDO (SARDINHA, CAVALA, SABEL)"
907 PRINT "19 - PEIXE MAGRO (PESCADA, FANECA, RAIA...)"
908 PRINT "20 - BACALHAU"
909 PRINT "21 - POLVO OU LULA"
910 PRINT "22 - PEIXE DE CONSERVA EM OLEO"
911 PRINT "23 - MARISCOS DE CONCHA"
912 PRINT "24 - CRUSTACEOS (LAGOSTA, CAMARAO, E SIMILARES)"
913 PRINT "GRUPO SEGUINTE: A"
914 GOTO 1000
915 CLS
916 PRINT "GRUPO A - AMILACEOS (CEREAIS...)"
917 PRINT "25 - PAO DE TRIGO"
918 PRINT "26 - MASSAS"
919 PRINT "27 - ARROZ"
920 PRINT "28 - BATATAS"
921 PRINT "29 - SEMENTES DE LEGUMINOSAS (FEIJAO, FAVA, GRAO-DE-BICO, ERVILHA, ETC.)"
922 PRINT "30 - SOJA"
923 PRINT "31 - LENTILHA..."
924 PRINT "32 - AMENDOIM"
925 PRINT "GRUPO SEGUINTE: H"
926 GOTO 1000
927 CLS
928 PRINT "GRUPO H - PRODUTOS HORTICOLAS"
929 PRINT "33 - COUVE, ALFACE, REPOLHO, AGRIAO, GRELOS, CHICORIA, E SIMILARE"
930 PRINT "34 - FEIJAO VERDE"
931 PRINT "35 - VAGEM DE ERVILHA"
932 PRINT "36 - CENOURA"
933 PRINT "37 - NABO"
934 PRINT "38 - CEBOLA"
935 PRINT "39 - TOMATE"
936 PRINT "40 - ABOBORA"
937 PRINT "GRUPO SEGUINTE: F"
938 GOTO 1000
939 CLS
940 PRINT "GRUPO F - FRUTAS"

```



```

941 PRINT "...", "41 - PERA, MACHA..."
942 PRINT "42 - PESSEGO, AMEIXA, CEREJA..."
943 PRINT "43 - LARANJA, TANGERINA..."
944 PRINT "44 - UVAS"
945 PRINT "45 - MORANGOS"
946 PRINT "46 - BANANAS"
947 PRINT "47 - CASTANHAS"
948 PRINT "48 - NOZES"
949 PRINT "...", "GRUPO SEGUINTE: D"
950 GOTO 1000
951 CLS
952 PRINT "GRUPO D - DOCES"
953 PRINT "...", "49 - ACUCAR"
954 PRINT "50 - MEL"
955 PRINT "51 - CHOCOLATE"
956 PRINT "52 - FRUTA PASSA"
957 PRINT "53 - BOLOS CASEIROS"
958 PRINT "54 - BISCOITOS..."
959 PRINT "55 - COMPOTAS"
960 PRINT "56 - MARMELADA"
961 PRINT "...", "GRUPO SEGUINTE: G"
962 GOTO 1000
963 CLS
965 PRINT "GRUPO G - GORDURAS"
966 PRINT "...", "57 - AZEITE"
967 PRINT "58 - OLEOS VEGETAIS"
968 PRINT "59 - MARGARINA"
970 PRINT "...", "ACABAM AQUI OS GRUPOS; MAS PODE"
971 PRINT "VOLTAR A ""CHAMAR"" QUALQUER UM..."
975 GOTO 1000
976 CLS
1000 INPUT A$
1002 IF A$="L" THEN GOTO 876
1004 IF A$="C" THEN GOTO 891
1006 IF A$="P" THEN GOTO 903
1008 IF A$="A" THEN GOTO 915
1010 IF A$="H" THEN GOTO 927
1012 IF A$="F" THEN GOTO 939
1014 IF A$="D" THEN GOTO 951
1016 IF A$="G" THEN GOTO 963
1018 IF A$="S" THEN GOTO 1100
1020 IF A$="R" THEN RUN
1030 LET N=VAL A$(1 TO 2)
1040 LET Q=VAL A$(3 TO )
1050 FOR M=1 TO 4
1060 LET B(N,M)=Q*A(N,M)/100
1070 NEXT M
1080 LET E(N)=B(N,1)*4.25+B(N,2)*9.5+B(N,3)*4.2
1090 GOTO 1000
1100 FAST
1102 LET P=0
1103 LET L=0
1104 LET G=0
1105 LET D=0
1106 LET E=0
1110 FOR N=1 TO 59
1120 LET P=P+B(N,1)
1130 LET L=L+B(N,2)
1140 LET G=G+B(N,3)
1150 LET D=(D+B(N,4))/10
1160 LET E=E+E(N)
1170 NEXT N
1175 LET P=INT (P/5+.5)*5
1180 LET L=INT (L/5+.5)*5
1185 LET G=INT (G/5+.5)*5
1190 LET D=INT (D/10+.5)*10
1195 LET E=INT (E/10+.5)*10

```



```

1200 LET PI=INT ((K*.12/4.25)/5)*5
1210 LET LI=INT ((K*.23/9.5)/5)*5
1220 LET GI=INT ((K*.65/4.2)/5)*5
1225 CLS
1300 PRINT "SUAS NECESSIDADES VALORES PRE-"
1301 PRINT "MEDIAS POR DIA, DA SUA CISA"
1302 PRINT "(EM CAL E GRAMAS) DIARIA DE"
1305 PRINT ", "ENERGIA (CAL)"
1307 PRINT ", "PROTIDOS (G)"
1309 PRINT ", "GORDURAS (G)"
1311 PRINT ", "GLICIDOS (G)"
1313 PRINT ", "DESPESAS DIARIAS:"
1320 PRINT ", "PODE TENTAR UM MELHOR ACERTO"
1322 PRINT ", "GRUPOS E ALIMENTOS MAIS RICOS EM"
1324 PRINT ", "PROTIDOS:C,P,L(4,5,6),A(29...32)"
1326 PRINT "GORDURAS:G,L(3,4,5,8),C(12)A(32)"
1327 PRINT "P(22),A(30),F(48),D(51)"
1328 PRINT "GLICIDOS:D,A,F"
1330 PRINT AT 4,14;K;AT 4,20;E;AT 4,28;K-E
1331 IF K>E THEN PRINT AT 4,27;"+"
1335 PRINT AT 6,14;PI;AT 6,20;P;AT 6,28;PI-P
1336 IF PI>P THEN PRINT AT 6,27;"+"
1340 PRINT AT 8,14;LI;AT 8,20;L;AT 8,28;LI-L
1341 IF LI>L THEN PRINT AT 8,27;"+"
1345 PRINT AT 10,14;GI;AT 10,20;G;AT 10,28;GI-G
1346 IF GI>G THEN PRINT AT 10,27;"+"
1350 PRINT AT 12,20;D;"$00"
1360 SLOW
1370 GOTO 1000

```

Sobre o programa
DIA DA SEMANA

(v. página seg.)

"O calendário actualmente utilizado pela maioria dos povos teve a sua origem no calendário romano, modificado por de terminação de Júlio César, no ano 46 a.C., com base em cálculos do astrónomo de Alexandria, Sozígemes. A modificação consistiu em acrescentar, de 4 em 4 anos, um dia ao ano civil de 365 dias. Como esse dia a mais se deveria intercalar a seguir ao sexto dia antes das Calendas de Março, daí chamar-se bissexto ao ano que houvesse de ter esse dia bis-sexto. Consistiu nisto a Reforma Juliana do calendário. Porém, com o decorrer dos anos verificou-se que tal reforma, que pretendia sincronizar o ano civil com o ano astronómico, não surtiu o efeito desejado. Assim, no séc. XVI o ano civil já se encontrava 10 dias adiantado em relação ao ano astronó-

mico. Foi essa a razão por que o papa Gregório XIII, em 1582, mandou "suprimir" 10 dias no calendário: do dia 4 de Outubro passou-se ao dia 15 de Outubro. E para que não se repetisse o desfaseamento entre o ano civil e o astronómico, determinou o mesmo papa que daí em diante os anos seculares não fossem bissextos, como deveriam ser segundo a reforma juliana (um ano é bissexto quando divisível por 4...) - excepto os anos seculares cuja centena fosse divisível por 4 (1600, 2000, 2400...): esses continuariam a ser bissextos. Portanto, não seriam bissextos 1700, 1800, 1900, 2100, etc... Com este recurso, o sincronismo entre o ano civil e o astronómico conservar-se-á durante milhares de anos. Consistiu isto na chamada Correção Gregoriana do calendário (...) e o calendário civil assim modificado denomina-se CALENDÁRIO GREGORIANO - que é o nosso calendário. (...) " J.F.G.

PROGRAMA "DIA DA SEMANA"

Autor: J. F. G. / Braga

```

1 REM "D/S"
2 PRINT "ESTE PROGRAMA PERMITE DETERMINAR"
3 PRINT "QUE DIA DA SEMANA FOI OU VAI SER"
4 PRINT "QUALQUER DATA DA ERA DE CRISTO."
5 PRINT "....."PARA ISSO, DEVEM-SE INTRODUIR,"
6 PRINT "SEPARADAMENTE, E POR ESTA ORDEM:"
7 PRINT ".....PRIMEIRO, O DIA DO MES;"
8 PRINT ".....DEPOIS, O NUMERO DO MES;"
9 PRINT ".....E, FINALMENTE, O ANO."
10 PRINT "....."QUANDO O COMPUTADOR SUGERIR QUE"
11 PRINT "INTRODUZA UMA DATA, INTRODUZA-A."
15 PAUSE 500
16 POKE 16437,255
18 CLS
20 PRINT "DATA ";
25 INPUT D
30 PRINT D;" / ";
35 LET I=NOT D
40 INPUT M
45 PRINT M;" / ";
50 INPUT A
55 PRINT A;" - ";
60 LET B=NOT A
65 LET R=A+INT (A/4)
70 LET S=INT (A/100)
75 LET C=INT ((S-16)*.75+.75)
80 IF A<1582OR (A=1582AND M<10)OR (A=1582AND M=10AND D<=4)THEN LET E=R+12
85 IF A>1582OR A=1582AND M>10OR A=1582AND M=10AND D>=15THEN LET E=R-(12+C)
90 LET F=INT (7*(E/7-INT (E/7))+.5)
100 IF A=4*INT (A/4)THEN LET B=A
105 IF A>1582AND A=S*100THEN LET B=NOT A
110 IF A>1582AND A=S*100AND A=400*INT (S/4)THEN LET B=A
115 IF M=1AND A=BTHEN LET N=6
120 IF M=2AND A=BTHEN LET N=2
125 IF M=1AND A<>BTHEN LET N=7
130 IF M=2AND A<>BTHEN LET N=3
135 IF M=3OR M=11THEN LET N=3
140 IF M=4OR M=7THEN LET N=6
145 IF M=5THEN LET N=1
150 IF M=6THEN LET N=4
155 IF M=8THEN LET N=2
160 IF M=9OR M=12THEN LET N=5
165 IF M=10THEN LET N=7
170 LET K=INT (((F+N+D)/7-INT ((F+N+D)/7))*7+.5)
175 IF D>31OR D>30AND (M=4OR M=6OR M=9OR M=11)OR D>29AND M=2AND A=BOR D>28AND M
=2AND A<>BOR (D>4AND D<15)AND M=10AND A=1582THEN LET I=D
180 IF I=DTHEN PRINT "INEXISTENTE"
185 IF I=DTHEN GOTO 20
190 IF K=0THEN PRINT "SABADO"
191 IF K=1THEN PRINT "DOMINGO"
192 IF K=2THEN PRINT "SEGUNDA-FEIRA"
193 IF K=3THEN PRINT "TERCA-FEIRA"
194 IF K=4THEN PRINT "QUARTA-FEIRA"
195 IF K=5THEN PRINT "QUINTA-FEIRA"
196 IF K=6THEN PRINT "SEXTA-FEIRA"
198 GOTO 20

```

"(...) Do anteriormente exposto se depreende que NÃO É TÃO SIMPLES COMO ISSO SABER EM QUE DIA DA SEMANA CAIU OU CAIRÁ QUALQUER DATA. MAS ESTE PROGRAMA PERMITE SABÊ-LO, E PARA TODA A ERA CRISTÃ, TANTO PARA DADAS ANTERIORES COMO PARA DATAS POSTERIORES À CORRECÇÃO GREGORIANA. UM PROGRAMA ÚTIL, PORTANTO, PARA INVESTIGADORES, HISTORIADORES... E CURIOSOS, COMO O AUTOR DESTA PROGRAMA."

J. F. G.
BRAGA

Este trabalho permite-lhe a elaboração do desenho que pretender, com formas bastante rigorosas. Desde o círculo à linha diagonal, podem ser representados nas dimensões que necessitar, (dentro dos limites do ecrã), nestes dois casos não com o rigor absoluto como se desejaria, estando no entanto a perfeição dependente do seu trabalho.

Ex:

Se pretende desenhar uma linha horizontal ou vertical:

- Coloque o cursor "■", onde preten de iniciar a linha.
- Pressione a tecla " D ", o que se rá assinalado por um Beep.
- Desloque o cursor até ao final do comprimento desejado.
- Pressione de novo a tecla " D ". (repare que no final o Beep é mais prolongado).

No caso de linhas diagonais, o problema é identico, apenas que o desenho da linha não segue o percurso do cursor, como é lógico. Limita-se a assinalar o intervã-lo entre as posições do cursor, quando assinaladas pela tecla " D ".

O círculo não necessita de todo este trabalho, basta pressionar a tecla " Z "; dar a medida do raio; "Enter".

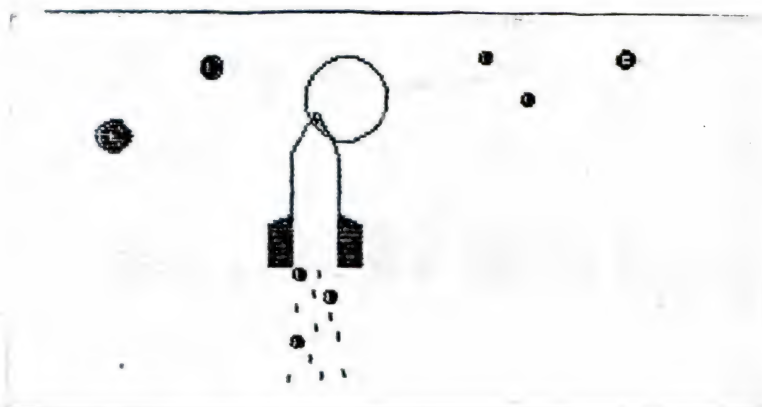
O deslocamento do cursor pode-se fazer de duas formas: em espaços curtos, da medida do cursor; em intervalos de 8 em 8, neste caso utilizando simultâneamente "SYMBOL SHIFT" e a tecla correspondente ao movimento que se propõe fazer.

Se tencionar apagar qualquer erro, deve deslocar o cursor em intervã-los curtos sobre a linha em que co meteu a falta.

Agora o desenho que elaborou pode ser gravado, para isso utilize a tecla " S "; dê o nome que desejar à figura; inicie o gravador; " Enter ".

Se pretender continuar ou relembrar o trabalho já gravado na cassette: passe de novo o programa para o seu SPECTRUM; "Run" e "Enter"; pressione a tecla " K "; passe a gravação do desenho. (Quando pressiona a tecla " S ", consegue apenas a gravação do desenho e não a do programa). Está pronto a utilizar o seu trabalho da forma que pretender; propomos como abertura de qualquer programa...

Está ao dispor da sua imaginação.



```

10 GO SUB 9000
15 GO TO 5000
20 LET FL=0: LET K=9: LET L=10
0: LET P=0
35 PRINT AT 1,6;"AUXILIAR DE D
ESENHO"
40 PRINT OVER 1;AT 2,0;"_____
45 PRINT AT 20,0;"_____

50 PLOT K,L
60 PRINT AT 21,15; INK 0;"POSI
CAO: ";K;",";L;","
100 LET A$=INKEY$: IF A$="" THE
N GO TO 50
104 IF A$="K" OR A$="k" THEN IN
PUT "Nome da figura."; LINE N$:
PRINT AT 21,0;"_____"
0,0;"_____"
";AT 0,6;n$: LOAD N$SCREEN$

105 IF P=0 THEN PLOT OVER 1;K,L
107 IF A$="S" OR A$="s" THEN PR
INT AT 21,0;"Nome da figura
": INPUT "(10 caracteres max)
"; LINE N$: PRINT AT 21,0;"
";AT 0,0;"_____"
";AT 0,10;n
$: FOR M=10 TO LEN N$: PRINT AT
0,M; OVER 1;"_";: NEXT M: SAVE N
$SCREEN$
108 IF A$="I" OR A$="L" THEN CL
S: GO TO 35
109 IF A$="1" OR A$="2" OR A$="
0" OR A$="3" OR A$="4" OR A$="5"
OR A$="6" OR A$="7" THEN INK VA
L A$: BORDER VAL A$: BEEP .01,30
110 IF (A$="U" OR A$="u") AND (
<151 THEN LET L=L+1
120 IF (A$="Y" OR A$="y") AND (
>=10 THEN LET L=L-1
130 IF (A$="I" OR A$="i") AND K
<=254 THEN LET K=K+1
140 IF (A$="T" OR A$="t") AND K
>1 THEN LET K=K-1
191 IF CODE A$=62 AND K>8 THEN
LET K=K-8
192 IF CODE A$=172 AND K<240 TH
EN LET K=K+8
193 IF CODE A$=197 AND L<140 TH
EN LET L=L+8
194 IF CODE A$=198 AND L>20 THE
N LET L=L-8
195 IF A$="H" OR A$="h" THEN GO
TO 5000
196 IF A$="C" OR A$="c" THEN IN
PUT "Nome da figura."; LINE N$:
PRINT AT 0,0;"_____"
";AT 21,0;"_____"
";AT 0,6;n$: COPY
197 IF A$="d" OR A$="D" THEN BE
EP .05,50: GO SUB 6000
198 IF A$="E" OR A$="e" THEN ST
OP

```



```

28 PRINT "As teclas para contr
ole do carro sao as seguintes:
(esquerda)
(para baixo)
(para cima)
(direita). Em caso de emergenci

```


pres-sionar a tecla 1, a qual dobra a velocidade do carro, fazendo-o, no entanto, gastar muito mais fuel.

```
27 PRINT "BOA CONDUCAO E BOA S
ORTE"
28 PAUSE 2300: CLS
29 FOR r=0 TO 7: POKE USR "a"+
r,60: NEXT r
30 FOR r=0 TO 7: POKE USR "b"+
r,0: IF r>1 AND r<6 THEN POKE US
R "b"+r,255
40 NEXT r
50 FOR i=146 TO 154
60 FOR r=0 TO 7
70 READ a: POKE USR CHR$(i)+r
```

```
80 NEXT r
90 NEXT i
100 DATA BIN 00011000,BIN 00011
000,BIN 01111110,BIN 01111110,25
5,255,255,BIN 11000011
110 DATA BIN 11000011,255,255,2
55,BIN 01111110,BIN 01111110,BIN
00011000,BIN 00011000
120 DATA BIN 11110000,BIN 11111
100,BIN 01111100,BIN 01111111,BI
N 01111111,BIN 01111100,BIN 1111
1100,BIN 11110000
130 DATA BIN 00001111,BIN 00111
111,BIN 00111110,BIN 11111110,BI
N 11111110,BIN 00111110,BIN 0011
1111,BIN 00001111
140 DATA BIN 00011000,BIN 00111
100,BIN 01111110,BIN 00011000,BI
N 00011000,BIN 00111100,BIN 1110
0111,BIN 11000011
150 DATA BIN 11000011,BIN 11100
111,BIN 00111100,BIN 00011000,BI
N 00011000,BIN 01111110,BIN 0011
1100,BIN 00011000
160 DATA BIN 11,BIN 00100011,BI
N 01100110,BIN 11111100,BIN 1111
1100,BIN 01100110,BIN 00100011,B
IN 11
170 DATA BIN 11000000,BIN 11000
100,BIN 01100110,BIN 00111111,BI
N 00111111,BIN 01100110,BIN 1100
0100,BIN 11000000
180 DATA 0,0,60,60,60,60,0,0
190 REM
191 REM MAIN PROGRAM
192 CLS
200 PRINT AT 0,11;"MISSEIS :";m
+1: FOR i=1 TO 21: PRINT AT i,0;
" ";AT i,30;" ";NEXT i
205 PRINT AT 1,0;" ";AT 21,0;" "
```

```
210 FOR i=2 TO 20: PRINT AT i,1
; INK 0;" ";NEXT i
220 PRINT AT 3,2;" ";AT 19,2;" "
```

```
230 FOR i=4 TO 9: PRINT AT i,2;
" ";AT i,28;" ";AT i+9,2;" ";AT
i+9,28;" ";NEXT i
240 PRINT AT 10,2;" ";AT 11,1;
INK 4;" ";INK 1;AT 12,2;" ";
250 PRINT AT 10,28;" ";AT 11,27
; INK 4;" ";INK 1;AT 12,28;" "
```

```
260 PRINT AT 3,14;" ";INK 4;"
";INK 1;" ";AT 19,14;" ";INK 4
";INK 1;" ";
270 PRINT INK 4;AT 2,15;" ";AT
4,15;" ";AT 18,15;" ";AT 20,15;" "
```

```
280 PRINT AT 5,4;" ";INK 4;"
";INK 1;" ";
AT 17,4;" ";INK 4;" ";
; INK 1;" ";
290 FOR i=6 TO 10: PRINT AT i,4
;" ";AT i,26;" ";AT i+6,4;" ";AT
i+6,26;" ";NEXT i
300 PRINT AT 10,4;" ";AT 12,4;"
";AT 10,26;" ";AT 12,26;" ";
310 PRINT INK 4;AT 11,4;" ";AT
11,26;" ";
320 PRINT AT 7,8;" ";INK 4;"
";INK 1;" ";AT 1
5,8;" ";INK 4;" ";INK
1;" "
```

```
330 FOR i=8 TO 9: PRINT AT i,6;
" ";AT i,24;" ";AT i+5,6;" ";AT
i+5,24;" ";NEXT i
340 PRINT AT 10,6;" ";AT 10,24;
" ";AT 12,6;" ";AT 12,24;" ";
350 PRINT INK 4;AT 11,6;" ";AT
11,23;" ";AT 6,15;" ";AT 8,15;
" ";AT 14,15;" ";AT 16,15;"
360 FOR i=8 TO 22 STEP 2: PRINT
AT 9,i;" ";AT 13,i;" ";NEXT i
370 FOR i=10 TO 12: FOR j=8 TO
22 STEP 2: PRINT AT i,j;" ";NEX
T j: NEXT i
```

```
374 REM
375 REM THE GAME
376 REM
380 INK 4
400 LET m=m+1: LET s=1: IF m=3
THEN LET m=2: LET s(2)=2
425 FOR o=1 TO s
435 LET o$=m$
436 IF INKEY$="1" THEN LET h$=0
$: LET g$="": LET g$=("1" AND h$
="0")+("0" AND h$="1"): LET s=2*
(h$="1")+ (h$="0")
437 IF s=2 THEN LET fuel=fuel-1
: IF fuel<=0 THEN LET g$="0": LE
T s=1
438 LET fuel=fuel-1: PRINT INK
2;AT 0,23;"FUEL:";fuel*(fuel>=0)
: IF fuel<0 THEN GO TO 700
440 IF INKEY$<>" " AND INKEY$<>"
1" THEN LET m$=INKEY$
441 LET oldy=y: LET oldx=x
445 PRINT AT y,x;" "
450 LET x=x+(m$="8" AND (ATTR (
y,x+1)=56 OR ATTR (y,x+1)=60))- (
m$="5" AND (ATTR (y,x-1)=56 OR A
TTR (y,x-1)=60))
460 LET y=y+(m$="6" AND (ATTR (
y+1,x)=56 OR ATTR (y+1,x)=60))- (
m$="7" AND (ATTR (y-1,x)=56 OR A
TTR (y-1,x)=60))
500 IF ATTR (y,x)=56 THEN BEEP
.001,40: LET score=score+10: LET
fuel=fuel+1: PRINT AT 0,0; INK
2;"SCORE:";score: LET dots=dots+
1: IF dots>=200 THEN LET s(1)=2:
IF dots>=290 THEN LET dots=0: L
ET s(1)=1: GO TO 10
504 IF (x=x(1) AND y=y(1)) OR (
x=x(2) AND y=y(2) AND m=2) THEN
GO TO 700
505 IF oldy=y AND oldx=x THEN L
ET m$=0$
510 IF m$="5" OR m$="6" OR m$="
7" OR m$="8" THEN LET d$="": LET
d$=("5" AND m$="6")+("6" AND m$
="7")+("7" AND m$="8")+("8" AND
m$="5")
515 PRINT AT y,x; INK 2;d$
516 NEXT o
520 FOR p=1 TO m
530 FOR q=1 TO s(p)
535 LET p$="": IF AND>.7 THEN G
O TO 570
540 LET p$=("8" AND x(x(p) AND
(ATTR (y(p),x(p)-1)=56 OR ATTR (
y(p),x(p)-1)=56 OR ATTR (y(p),x(
p)-1)=60))+("5" AND x>x(p) AND (
ATTR (y(p),x(p)+1)=56 OR ATTR (y
(p),x(p)+1)=56 OR ATTR (y(p),x(p
)+1)=60))
550 IF p$<>" " THEN GO TO 580
570 LET p$=("7" AND y(p)>y AND
(ATTR (y(p)-1,x(p))=56 OR ATTR (
y(p)-1,x(p))=56 OR ATTR (y(p)-1,
x(p))=60))+("8" AND y(p)<y AND (
ATTR (y(p)+1,x(p))=56 OR ATTR (y
(p)+1,x(p))=56 OR ATTR (y(p)+1,x
(p))=60))
575 IF p$=" " THEN LET p$=e$(p):
GO TO 577
580 PRINT AT y(p),x(p);" ": IF
o(p)=56 THEN PRINT AT y(p),x(p);
INK 0;" "
590 LET x(p)=x(p)-(p$="8" AND (
ATTR (y(p),x(p)-1)=56 OR ATTR (y
(p),x(p)-1)=60 OR ATTR (y(p),x(p
)-1)=56))+ (p$="5" AND (ATTR (y(p
),x(p)+1)=56 OR ATTR (y(p),x(p)+
1)=60 OR ATTR (y(p),x(p)+1)=56))
: IF p$="7" OR p$="6" THEN LET y
(p)=y(p)-(p$="7" AND (ATTR (y(p)
-1,x(p))=56 OR ATTR (y(p)-1,x(p)
)=56 OR ATTR (y(p)-1,x(p))=60))+
```



```

(p$="6" AND (ATTR (y(p)+1,x(p))=
56 OR ATTR (y(p)+1,x(p))=56 OR A
TTR (y(p)+1,x(p))=60))
600 LET o(p)=ATTR (y(p),x(p))
610 IF p$("<") THEN LET f$(p)="
: LET f$(p)="$" AND p$="7")+("$
" AND p$="6")+("$K" AND p$="8")+
"$" AND p$="5")
620 PRINT AT y(p),x(p);f$(p)
625 LET e$(p)=p$
628 IF y(p)=y AND x(p)=x THEN G
O TO 700
630 NEXT q: NEXT p
640 GO TO 425
700 INK 2: BRIGHT 1: FLASH 1: P
APER 7: CLS: BEEP 1,-40: FLASH
0: BRIGHT 0: CLS: PRINT AT 10,1
0:"SCORE:";score
710 PRINT AT 20,0;"AZAR...PERDE
U": PAUSE 200
720 GO TO 19

```

```

1100 RETURN
2000 DATA 0,2,10000,0,INT (1000*
AND),0,2000,"ESQUERDA","VERTICAL
","DIREITA"
2010 RESTORE 2000: DIM d$(3,6):
READ j,d,h,v,hp,hv,f,d$(1),d$(2)
,d$(3)
2020 LET b$="....."
.....": LET c$=b$+b$: LE
T b=1: RETURN
3000 DIM r$(8): DATA STR$ j,d$(d
),STR$ v,STR$ h,STR$ hv,STR$ hp,
STR$ f
3010 RESTORE 3000: FOR n=1 TO 7:
READ r$: PRINT AT 1+2*n,16; INK
7; BRIGHT 1: PAPER 2-(n>2); FLA
SH ((n=3)*(ABS v>200)+(n=4)*(h<5
00)+(n=5)*(ABS hv>100)+(n=6)*(AB
S hp>1000)+(n=7)*(f(100)));r$: NE
XT n
3020 LET b=INT (hp/10+.5): LET b
=b-32*INT ((b-1)/32): PRINT AT 1
8,0; INK 4;c$(b TO b+31)
3030 LET h1=INT (16-h/1000): LET
h1=h1+(h1<1)*(-h1): PRINT AT h
1,27;"":AT h1+3,27;"":AT h1+
1,27; INK 5;a$(d,1);AT h1+2,27,
$(d,2)
3040 FOR m=0 TO 1: FOR n=0 TO 8:
STEP 4: FOR o=0 TO 1: PLOT INVER
SE m; INK 2; BRIGHT 1;231+o+n-6*
4,151-h1*8: DRAW INK 2; INVERSE
m;2+j*(d=1)-(d=3))-2+j: NEXT n
: NEXT m
3050 RETURN
4000 BORDER 3: INK 1: PAPER 6: C
LS
4010 BRIGHT 1: PRINT PAPER 1; IN
K 7;"
4020 PRINT "A sua missao e po
isar o modulo Lunar, a salvo e n
o lugar certo,que aparece assina
lado."
4030 PRINT "Voce pode alterar
a direccao do modulo com a tecla
0 para esquerda e com a P para
a direita, ou a potencia dos jac
tos, com as teclas 1 a 9."
4050 PRINT "Quer a velocidade
vertical ou horizontal,nao pod
e ser superior a 10 m/s, no momen
to de poiso."
4060 PRINT "O correcto local do
poiso tem um raio maximo de
metros"
4070 PRINT "A tecla actuada e
sta incapacitada, ate ao apareci
mento dum quadrado amarelo no t
opo esquerdo do ecran."
4080 RETURN
5000 DATA 15,127,123,119,239,223
,255,255
5010 DATA 0,128,192,192,192,192,
192,128
5020 DATA 127,62,0,0,0,0,0,0
5030 DATA 255,255,196,196,192,24
0,224,192
5040 DATA 3,31,56,63,63,63,31,7
5050 DATA 192,248,60,252,252,252
,248,224
5060 DATA 3,7,14,28,56,124,0,0
5070 DATA 192,224,112,56,28,62,0
,0
5080 DATA 0,1,3,3,3,3,3,1
5090 DATA 240,254,222,238,247,25
1,255,255
5100 DATA 255,255,99,35,3,15,7,3
5110 DATA 254,124,0,0,0,0,0,0
5120 DIM a$(3,2,2): FOR a=0 TO 1:
1: RESTORE 5000+10*a: FOR n=USR
CHR$ (97+a) TO USR CHR$ (97+a)+7
: READ b: POKE n,b: NEXT n: NEXT
a
5130 FOR a=1 TO 3: FOR b=1 TO 2:
FOR c=1 TO 2: LET a$(a,b,c)=CHR
$ (137+4*a+2*b+c): NEXT c: NEXT
b: NEXT a
5140 RETURN

```

SPECTRUM

Programa "ALUNAGEM"

Autor:

FERNANDO PRECES

```

100 GO SUB 4000: GO SUB 5000: I
NPUT "Prima ENTER para comecar
: LINE z$: GO SUB 2000: GO SUB 1
000: GO SUB 3000
110 LET z$="": PAUSE 1: LET z$=
INKEY$
115 PRINT AT 2,0;"": IF z$("<")
THEN PRINT AT 2,0; PAPER 6;"
120 IF (z$="p") AND (d<3) THEN
LET d=d+1
130 IF (z$="q") AND (d>1) THEN
LET d=d-1
140 IF (z$>="0") AND (z$<="9")
THEN LET z=VAL z$: LET j=z
145 LET j=j*(f>0)
150 LET v=INT (.5+v+(.7+.3*(d=2
)))*j-5+(ABS hv+2)/5000)
160 LET h=INT (h+v+.5): LET h=h
*(h>0)
170 LET hv=INT (.5+hv+.7*j*((d=
3)-(d=1)))
180 LET hp=((5*hp+hv)/5)
190 LET f=((f>0)*INT (.5+f-j)
200 GO SUB 3000
210 IF f=0 THEN PRINT AT 1,16;
PAPER 2; FLASH 1;" FUEL OFF "
220 IF (h<10) AND (ABS v<10) AN
D (ABS hp<10) AND (ABS hv<10) TH
EN PRINT AT 1,0; PAPER 4; INK 0;
FLASH 1;" BOA ALUNAGEM ": INPUT
"Prima ENTER para jogar outra v
ez": LINE z$: RUN
230 IF (h<10) AND (ABS v<10) AN
D (ABS hv<10) THEN PRINT AT 1,0;
INK 0; PAPER 4; FLASH 1;" BOA A
LUNAGEM ", PAPER 2; FLASH 1;"Em
local errado ": INPUT "Prima ENT
ER para jogar outra vez": LINE z
$: RUN
240 IF h<10 THEN PRINT AT 1,0;
PAPER 2; FLASH 1;" ACIDENTE - CH
OGUE VIOLENTO ": INPUT "Prima EN
TER para outro jogo": LINE z$: R
UN
250 GO TO 110
1000 BORDER 0: INK 7: PAPER 0: C
LS
1010 PRINT PAPER 1;"
ALUNAGEM
1020 PAPER 7: INK 2: PRINT "
JACTOS:"
1030 PRINT " DIRECCAO:"
1040 INK 1: PRINT " VEL VERTIC
AL:"
1050 PRINT " ALTURA:"
1060 PRINT "VEL HORIZONTAL:"
1070 PRINT "POS HORIZONTAL:"
1080 PRINT " FUEL:"
1090 PAPER 0: INK 7: PRINT PAPER
4;AT 19,0;

```


PROGRAMA PARA COMPUTADOR NEW BRAIN

ISABEL CRISTINA/Porto

```

10 OPEN#0,0,"100"
20 OPEN#1,11,"N160"
30 PLOT BACKGROUND (1),WIPE
40 PLOT RANGE (200,100)
50 PLOT CENTRE (0,0)
60 PLOT PLACE (10,10)
70 PLOT MOVE(10,90),MOVE(190,90)
80 PLOT MOVE (190,10),MOVE(10,10)
90 PLOT PLACE(70,50)
100 PLOT "GRAFICO"
110 PUT22,1,4:INPUT("PARA ATRIBUIR VALORES ACCIONE NEW LINE")T$:CLOSE#1:GOTO120
120 PUT31,7:"INTRODUZIR CADA VALOR ESCOLHIDO(NEW LINE Para terminar)":7
125 ?TAB(8):"PONTO=X":TAB(23):"PONTO=Y":N=1
130 PUT22,10,N+6:INPUT("X")XX$
135 IF XX$=""THEN145
137 PUT22,25,N+6:INPUT ("Y")YY$
140 X(N)=VAL(XX$):Y(N)=VAL(YY$):N=N+1:GOTO 130
145 N=N-1
150 PUT31,7:"PONTO"," X"," Y"
155 FOR VV=1 TO N:VV,X(VV),Y(VV):NEXT VV
157 PUT22,1,22:INPUT("PARA OBSERVAR O GRAFICO ACCIONE NEW LINE")T$
160 OPEN#0,0,"100":OPEN#10,11,"n180"
165 PLOT RANGE(200,100),CENTRE(20,20)
170 PLOT PLACE(0,0),AXES(10,10)
175 FOR I =1 TO N
180 PLOT MOVE(X(I),Y(I)):NEXT I
185 PUT22,4,5:?"Y-EIXO":Y1:":":Y2:TAB(5):"X-EIXO":X1:":":X2:7
190 PUT22,4,2:INPUT("NEW LINE PARA NOVOS VALORES")D$:CLOSE#10:GOTO120
200 END

```

Este programa executa o traçado de uma recta, relativamente aos valores que lhe são atribuídos.

Para iniciar o programa, actue o comando RUN.

Execute depois as instruções que lhe vão sendo indicadas.

CARACTERIZAÇÃO DO GRÁFICO

. Na linha 160, encontra-se a instrução para abrir o ecrã de gráficos; "n180" dimensiona o comprimento do eixo Y.

. Na linha 165, encontra-se o posicionamento dos eixos X e Y. Poderemos, por exemplo, deslocar o eixo Y ao longo do eixo X, alterando os valores de "CENTRE".

. Na linha 170, encontra-se a escala dos eixos que, neste caso, é de 1 por 1; poderá ser alterada, modificando os valores de "AXES".

. Na linha 180, encontra-se a instrução de posicionamento dos valores introduzidos.

Este programa também executa rectas, com valores negativos.

INTRODUÇÃO À PROGRAMAÇÃO ESTRUTURADA

(contin. do nº anterior)

ARBORESCÊNCIAS

Já vimos que as listas lineares são estruturas dinâmicas, análogas, sob certos pontos de vista, aos vectores estatísticos. A única diferença residia na capacidade de fazer variar a dimensão e a disposição da estrutura.

Tais listas apenas produzem uma ordenação dos elementos e não permitem construir uma organização hierárquica dos elementos.

Inversamente, as arborescências são muito utilizadas na informática. Uma arborescência (diz-se também árvore) é um conjunto de elementos organizados de modo hierárquico. As árvores em informática crescem para baixo - diz-se que a raiz da arborescência está em cima, os ramos estendem-se para baixo, sendo as folhas os elementos terminais, ou seja, os mais baixos da árvore.

As arborescências permitem a representação de um grande número de situações e de fenómenos: decomposição de um programa em sub-programas, árvore de avaliação de um jogo de estratégia (xadrez, damas, etc.), sintaxe de uma expressão aritmética (fig. 8) ou de uma linguagem de programação, classificações diversas, tal como uma classificação zoológica em espécies, géneros, variedades, etc. (fig. 9).

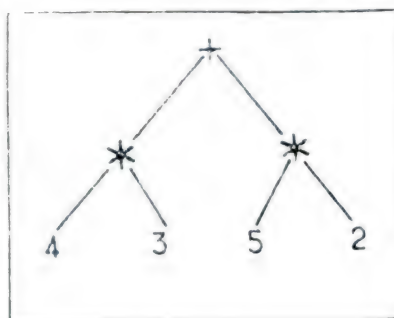


Fig. 8
As arborescências podem representar a sintaxe de uma expressão aritmética.

As arborescências são também utilizadas em muitos domínios da informática: compilação, concepção de sistemas de exploração, Inteligência Artificial, construções de bases de dados, etc.

Assinale-se que uma árvore se define formalmente do seguinte modo: denomina-se árvore de tipo T uma estrutura de dados do mesmo tipo, que se designa raiz, e de uma sequência de árvores do mesmo tipo, que se designam sub-árvores; esta sequência pode ser nula. Em comparação com uma árvore genealógica, chama-se "nós-filhos" aos nós provenientes da raiz, e "nó-pai" à raiz de uma sub-árvore.

As arborescências não tem uma representação física própria. Para podermos implantar esta estrutura, vamos analisar uma arborescência de um tipo particular: a árvore binária, que se representa directamen-

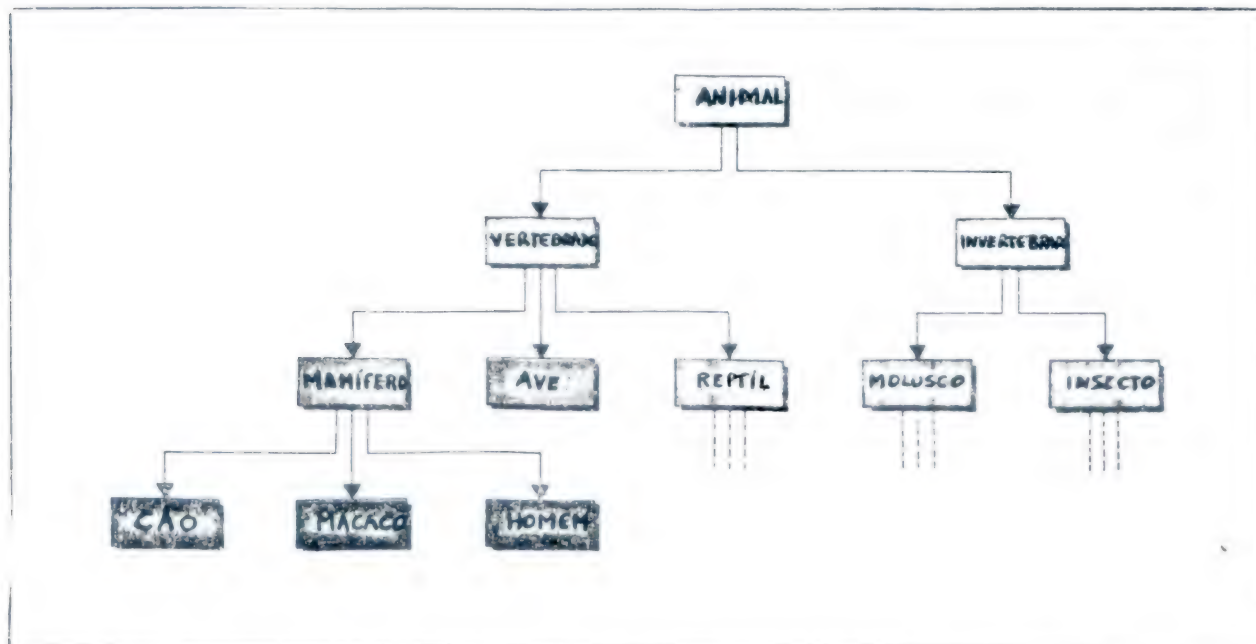
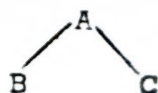


Fig. 9 - Uma classificação zoológica que respeita uma ordem hierárquica pode ser formulada em forma de uma arborescência.

te na máquina. Além disso, veremos que qualquer árvore pode ser reproduzida numa árvore binária.

Uma árvore binária é uma árvore cujos nós apenas possuem duas ramificações, e para a qual se distingue linhas-esquerda e linhas-direita. Dito de outro modo uma árvore da forma



será diferente de



Entanto, esta restrição é uma vantagem pois, ao tratarmos quaisquer arborescências, damos um carácter diferente às ramificações conforme estejam à direita ou à esquerda.

Uma árvore binária define-se logicamente pelas seguintes operações:

. ACESSO que se subdivide em três funções:

- raiz, que lê a raiz de uma árvore,
- direita, que lê a ramificação direita de uma árvore
- esquerda, que dá acesso à sub-árvore esquerda

. CONSTRUÇÃO: criação de uma árvore binária a partir de duas sub-árvores e de uma raiz

. TESTE: função vazia que determina se a sub-árvore está vazia ou não.

Há duas representações físicas possíveis de uma árvore binária: utilização de tabelas ou de agregados.

Implantada em forma de tabelas, a estrutura de uma árvore binária reduz-se a tres vectores. O vector dos valores, que contém a componente significativa de um nó, o vector dos ponteiros sobre as linhas esquerdas e o vector dos ponteiros sobre as linhas direitas.

A outra forma, que emprega a noção de agregado, é muito utilizada em Pascal.

Uma característica importante das árvores é a de poderem ser "percorridas", ou seja, é possível substituir o comprimento desta arborescência numa certa ordem e tratar os valores dos nós durante este percurso.

Os tres percursos principais que se podem realizar numa árvore denominam-se

programa expressão;

type arvore = ↑ elemento;

elemento = record

valor : car;

lnhe : arvore;

lnhd : arvore;

end;

var expr : arvore;

function criar(v:car;le,ld:arvore):arvore;

var p:arvore;

begin

new(p);

p↑.valor:=v;

p↑.lnhe:=le;

p↑.lnhd:=ld;

criar:=p;

end;

procedure preordre(a:arvore);

begin

if a nil then

begin

write(a↑.valor);

preordre(a↑.lnhe);

preordre(a↑.lnhd);

end;

end;

procedure postordre (a:arvore);

begin

if a<>nil then

begin

postordre(a↑.lnhe);

postordre(a↑.lnhd);

write(a↑.valor);

end;

end;

procedure inordre (a:arvore);

begin

if a<>nil then

begin

inordre(a↑.lnhe);

write(a↑.valor);

inordre(a↑.lnhd);

end;

end;

begin (X programa principal X)

expr:= criar('+',criar ('X',

criar('4',nil,nil),

criar('3',nil,nil)),

criar ('X',

criar('5',nil,nil),

criar('2',nil,nil)));

preordre(expr);

writeln;

postordre(expr);

writeln;

inordre(expr);

end.

Fig.10 - Criação e percurso de uma expressão aritmética representada em forma de uma árvore binária. Os percursos preordre, postordre e inordre dão respectivamente:
 "+*43*52", "43*52*+" e "4*3+5*2."

"préordem", "inordem" e "postordem" e exprimem-se de um modo muito simples:

"préordem": tratar primeiro a raiz e depois as linhas (filas).

"postordem": tratar primeiro as linhas e depois a raiz.

"inordem": tratar a linha esquerda, depois a raiz, depois a linha direita.

Percorrendo sucessivamente uma arborescência de expressão aritmética através de "préordem", "postordem" e "inordem", obtém-se as notações:

. "préordem": + * 4 3 * 5 2
(pré-fixado)

. "postordem": 4 3 * 5 2 +
(post-fixado)

. "inordem": 4 * 3 + 5 * 2
(infixado)

A figura 10 mostra uma implantação destes procedimentos em Pascal. Apresenta a construção de uma arborescência e o seu percurso por estes três algoritmos.

As árvores binárias, enquanto tais, tem numerosas aplicações. Uma delas, bastante útil, permite seleccionar os elementos, criando uma árvore binária de pesquisa.

O algoritmo desta triagem equivale a criar uma arborescência, inserindo sistematicamente os números inferiores à raiz na sub-árvore da esquerda, e os números superiores à raiz na sub-árvore da direita. (fig. 11)

Imaginemos que a sequência dos números a seleccionar é: 12 6 8 92 9 10 36 16 45 108 7 ... Após inserção destes números na arborescência, obtém-se a estrutura representada na figura 12. Depois, basta introduzir-lhe um percurso "inordem" para recuperar os números seleccionados. Este tipo de algoritmo de triagem, surpreendentemente, é muito rápido e bastante utilizado em triagens de tabelas.

```

programa inserção;
type arvore = ↑ elemento;
    elemento = record
        valor : inteiro;
        lnhe,lnhd : arvore;
    end;
var arv : arvore; n : inteiro;

function inserir ( x:inteiro;a : arvore): arvore;
begin
    if a = nil then
        inserir := criar (x,nil,nil)
    else
        if x <= a↑.valor
        then inserir := inserir(x,a↑.linhe)
        else inserir := inserir(x,a↑.linhd);
    end;
begin
    arv:=nil;
    repeat
        read(n);
        inserir(n,arv);
    until n= 0;
    inorder ( arv);
end.

```

Fig. 11 - Programa de triagem por inserção escrita em Pascal. As rotinas "criar" e "inorder" devem ser retomadas da fig. 10.

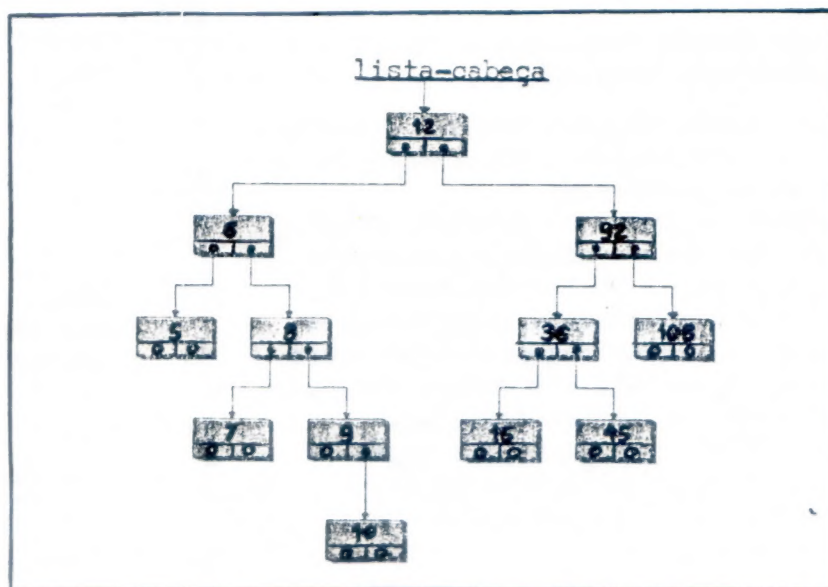


Fig. 12 - O estado da arborescência após inserção dos números da sequência 12 6 8 92 5 10 36 16 45 108 7. Um simples percurso "inorder" é suficiente para os re-ler classificados.

